University of Central Florida

Department of Computer Science

CDA 5106: Spring 2020

Machine Problem 1: Cache Design, Memory Hierarchy Design

by

<< Ali Almalki >>

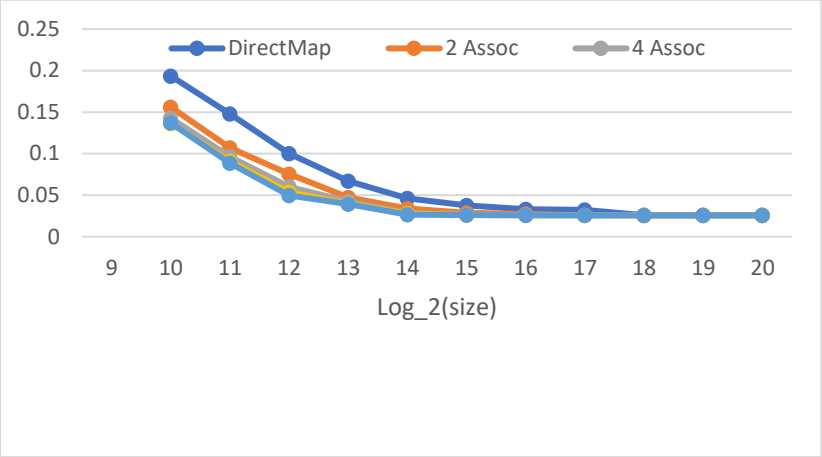
Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's electronic signature: Ali Almalki

(sign by typing your name)

Section 8.1. L1 Cache exploration: SIZE and ASSOC

Graph 1
L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
L2 cache: None
Prefetching: None



log_2(Size)

Discussion

In each of the associativity, with an increase in the cache size directly decrease miss rate.
For each cache size, incremental in the associativity decrements the miss rate with diminishing returns, i.e. 2 ASSOC is much better than Direct Map, but Fully ASSOC is not much better than 8 ASSOC.

Analyzing graph and data, the compulsory miss rate appears to be around 0.02582, ie. where the graphs bottom out.

Conflict miss rate can be measured by Direct Map Miss Rate – Fully ASSOC Miss Rate.

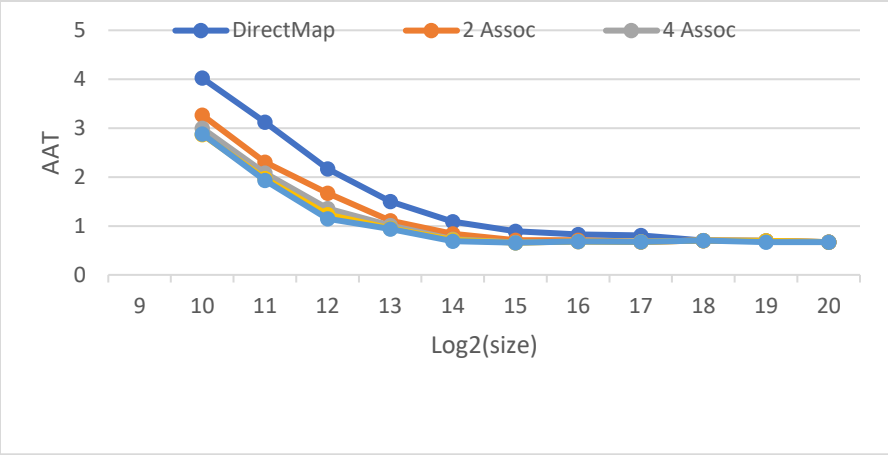
Graph2

Same as GRAPH #1, but the y-axis should be AAT instead of L1 miss rate.

Formula Used:

Miss_Penalty (in ns) = 20 ns + 0.1*(BLOCKSIZE / 16 B/ns)
= 20 + 0.1*(32/16) = 20.2 ns

Average access time (AAT) = HT_L1 + MR_L1* Miss_Penalty



Discussion

Analyzing the graph and data, associativity can be seen, 1024 kb cache has the best AAT, at 0.668569 ns.

Section 8.2. Replacement policy study

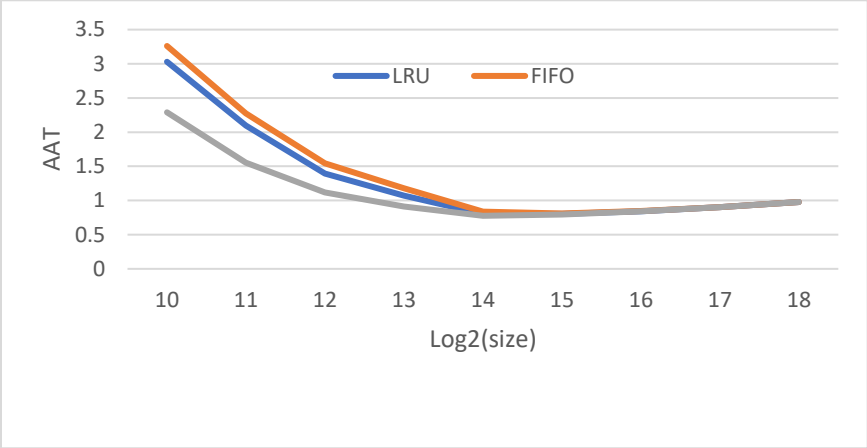
Graph 3

For this experiment: L1 cache: SIZE is varied, ASSOC = 4, BLOCKSIZE = 32. L2 cache: None. Replacement policy: varied Inclusion property: non-inclusive Plot AAT on the y-axis versus log2(SIZE) on the x-axis, for nine different cache sizes: SIZE = 1KB, 2KB, … , 256KB, in powers-of-two. (That is, log2(SIZE) = 10, 11, …, 18.) The graph should contain three separate curves (i.e., lines connecting points), one for each of the replacement policies: LRU, FIFO, optimal. All points for LRU replacement policy should be connected with a line, all points for FIFO replacement policy should be connected with a line, etc.

Formula Used:

Miss_Penalty (in ns) = 20 ns + 0.1*(BLOCKSIZE / 16 B/ns)
= 20 + 0.1*(32/16) = 20.2 ns

Average access time (AAT) = HT_L1 + MR_L1* Miss_Penalty



Discussion

Analyzing the data and graph optimal has the lowest AAT.

Section 8.3. Inclusion property study

Graph 4

For this experiment: L1 cache: SIZE = 1KB, ASSOC = 4, BLOCKSIZE = 32. L2 cache: SIZE = 2KB – 64KB, ASSOC = 8, BLOCKSIZE = 32. Replacement policy: LRU Inclusion property: varied

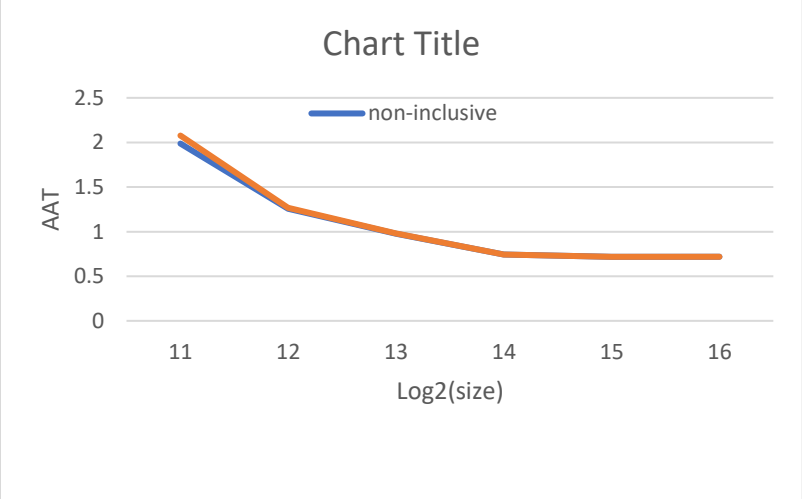
14

Plot AAT on the y-axis versus log2(L2 SIZE) on the x-axis, for six different L2 cache sizes: L2 SIZE = 2KB, 4KB, … , 64KB, in powers-of-two. (That is, log2(L2 SIZE) = 11, 12, …, 16.) The graph should contain three separate curves (i.e., lines connecting points), one for each of the following inclusion properties: non-inclusive and inclusive. All points for non-inclusive cache should be connected with a line, all points for inclusive cache should be connected with a line.

Miss_Penalty (in ns) = 20 ns + 0.1*(BLOCKSIZE / 16 B/ns)

= 20 + 0.1*(32/16) = 20.2 ns

AAT = HT_L1 + HT_L2*MR_L1 + MR_L2*MissPenalty



Discussion

Analyzing the graph and data non-inclusive properly yields a better AAT.